**SR UNIVERSITY**

**AI ASSISTED LAB**

**ASSIGNMENT 9.4**

Name: MOHAMMED MUTAKABIR HUSSAIN

H.no:2503A51L35

**TASK 1:**

**Task Description #1** (Automatic Code Commenting)

**Scenario:** You have been given a Python function without comments.

def calculate\_discount(price, discount\_rate):

return price - (price \* discount\_rate / 100)

• Use an AI tool (or manually simulate it) to generate line-by-line  
comments for the function.  
• Modify the function so that it includes a docstring in Google-style  
or NumPy-style format.  
• Compare the auto-generated comments with your manually  
written version

COMMENTS ADDED MANUALLY:

A screenshot of a computer

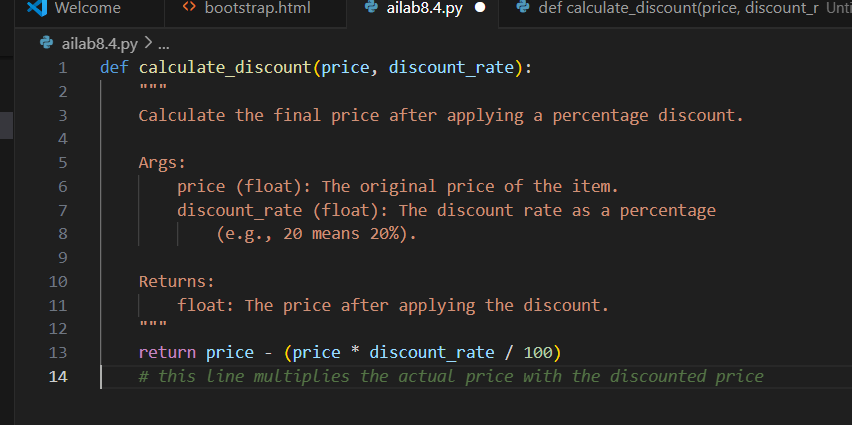
AI-generated content may be incorrect.

COMMENTS ADDED BY AI OUTPUT:

A screen shot of a computer

AI-generated content may be incorrect.

INCLUDING docstring in Google-style or NumPy-style format:



OBSERAVTION:  
I have observed how comments and docstrings help us to improve the readability of thee code and when compairing comments which were manually written and AI generated are how well the ai Is able to explain us showing in a clear format when comments were written manually we can write short and easy but takes a lot of time when the code is large , the use of docstring in google style or NumPy-style format made the code explainable to future developers

**TASK 2:**

**Task Description #2** (API Documentation Generator)

**Scenario:** A team is building a **Library Management System** with multiple functions.

def add\_book(title, author, year):

# code to add book

Pass

def issue\_book(book\_id, user\_id):

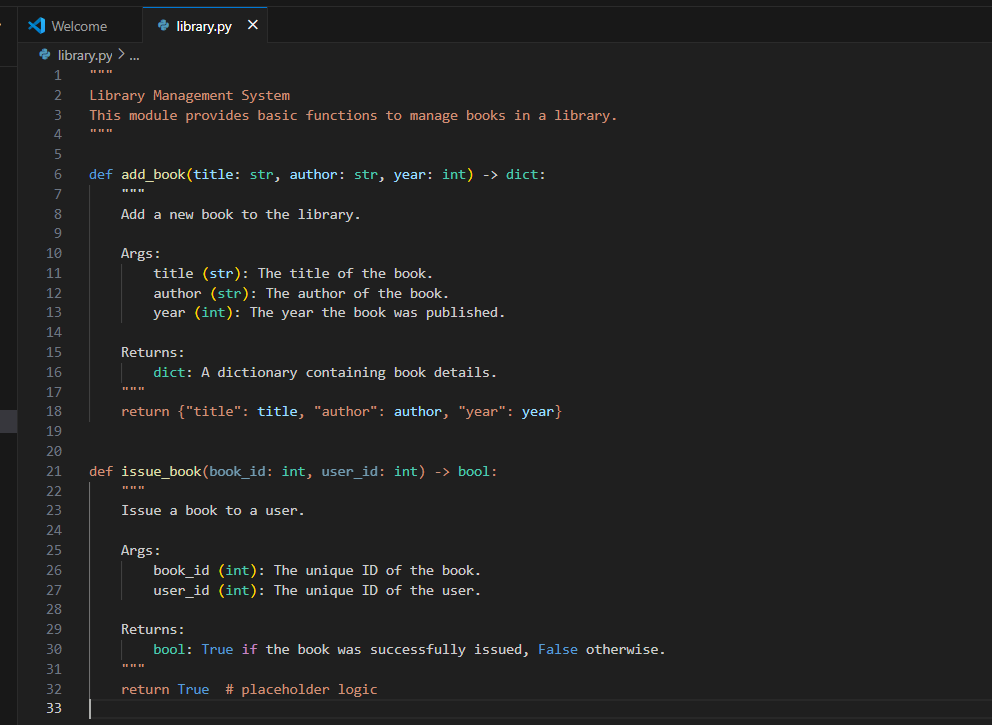
# code to issue book

Pass

• Write a Python script that uses docstrings for each function (with  
input, output, and description).  
• Use a documentation generator tool (like pdoc, Sphinx, or  
MkDocs) to automatically create HTML documentation.  
• Submit both the code and the generated documentation as output.

Prompt: Write a Python script that uses docstrings for each function (with  
input, output, and description).

Python script generated:



Installation of pdoc

documentation generator tool:

A screen shot of a computer

AI-generated content may be incorrect.



Output :

A screenshot of a computer

AI-generated content may be incorrect.

Observation;

**TASK 3**

**Task Description #3** (AI-Assisted Code Summarization)

**Scenario:** You are reviewing a colleague’s codebase containing long functions.

def process\_sensor\_data(data):

cleaned = [x for x in data if x is not None]

avg = sum(cleaned)/len(cleaned)

anomalies = [x for x in cleaned if abs(x - avg) > 10]

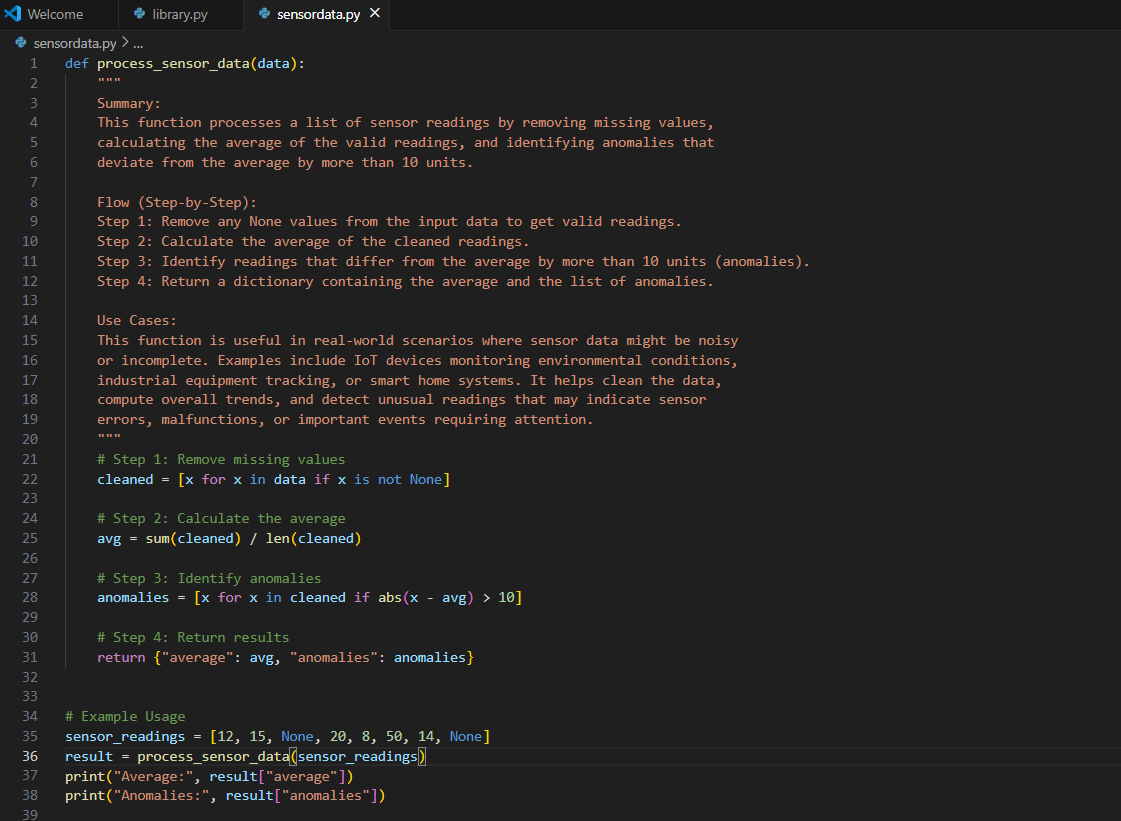
return {"average": avg, "anomalies": anomalies}

* Generate a summary comment explaining the purpose of the function in 2–3 lines.
* Create a flow-style comment (step-by-step explanation).
* Write a short paragraph of documentation describing possible use cases of this function in real-world scenarios.

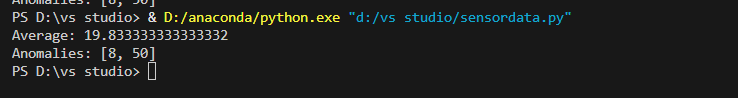
**Prompt:**

Generate a summary comment explaining the purpose of the function in 2–3 lines. Create a flow-style comment (step-by-step explanation). Generate summary comment and flow style comment and the example usage in the code itself

**Code generated :**



Output ;



**Observation:**  
This function cleans a list of sensor readings, finds the average, and shows any unusual values. It helps detect problems or unusual data, but it may give an error if the list is empty.

**TASK 4**

**Task Description #4** (Real-Time Project Documentation)

**Scenario:** You are part of a project team that develops a Chatbot Application. The team needs documentation for maintainability.

* Write a README.md file for the chatbot project (include project description, installation steps, usage, and example).
* Add inline comments in the chatbot’s main Python script (focus on explaining logic, not trivial code).
* Use an AI-assisted tool (or simulate it) to generate a usage guide in plain English from your code comments.

Reflect: How does automated documentation help in real-time projects compared to manual documentation

**Prompt:**

Make a simple Python chatbot that answers greetings and a few questions like how are you/hi or

It should keep talking until the user types exit

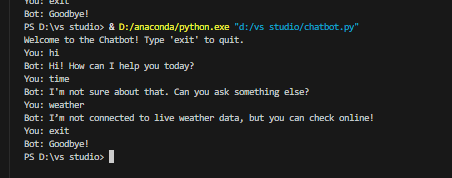
Add simple comments to explain what each part does for a beginner.

**Code:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Output:**



**Observation:**  
The chatbot works well for greetings and basic questions, responding correctly to things like hello or how are you but it cannot provide live information like the current time or weather because it is not connected to real-time data For anything it doesn’t understand, it gives a default response as I’m not sure about that. Can you ask something else?